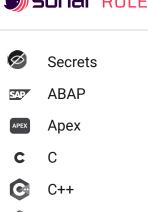
Code Smell (61)

Search by name...





CloudFormation

COBOL COBOL

C# C#

♥ CSS★ Flex

-**co** Go

**5** HTML

🎒 Java

JS JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I PL/I

PL/SQL PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB VB.NET

VB6 VB6

XML XML



## Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

Rug Bug

9

Tags

Security Hotspot (1)

All rules (76) 6 Vulnerability **(5**) Security.allowDomain(...) should only be used in a tightly focused manner Vulnerability flash.system.Security.exactSettings property should never be set to false Vulnerability Dynamic classes should not be used Code Smell "LocalConnection" should be configured to narrowly specify the domains with which local connections to other Flex application are allowed Vulnerability "default" clauses should be first or last Code Smell Event types should be defined in metadata tags Code Smell Event names should not be hardcoded in event listeners Code Smell The special "star" type should not be used Code Smell Variables of the "Object" type should Code Smell

Methods should not be empty

Constant names should comply with a

All branches in a conditional structure should not have exactly the same

Classes that extend "Event" should

Code Smell

naming convention

Code Smell

implementation

📆 Bug

Useless "if(true) {...}" and Analyze your code "if(false){...}" blocks should be removed cwe <table-of-contents> Bug 🔷 Major 🕝 if statements with conditions that are always false have the effect of making blocks of code non-functional. if statements with conditions that are always true are completely redundant, and make the code less readable. There are three possible causes for the presence of such code: • An if statement was changed during debugging and that debug code has been committed. • Some value was left unset. • Some logic is not doing what the programmer thought it did. In any of these cases, unconditional if statements should be removed. **Noncompliant Code Example** if (true) { doSomething(); } if (false) { doSomethingElse(); **Compliant Solution** doSomething(); See • MITRE, CWE-489 - Leftover Debug Code • MITRE, CWE-570 - Expression is Always False • MITRE, CWE-571 - Expression is Always True Available In: sonarcloud 🔂 | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy

override "Event.clone()"  • Bug
Constructors should not dispatch events
Rug
"ManagedEvents" tags should have companion "Event" tags
<b>∰</b> Bug
Objects should not be instantiated inside a loop
Two branches in a conditional structure should not have exactly the same implementation
Constructor bodies should be as lightweight as possible
Only "while", "do" and "for" statements should be labelled
Statements, operators and keywords specific to ActionScript 2 should not be used
"for" loop stop conditions should be invariant
Unused function parameters should be removed
Code Smell